**6 Строки. Классы String и StringBuilder**

Задание 1. Дан фрагмент текста, запрашиваемый у пользователя. Фрагмент содержит запись арифметического выражения. Написать программу, проверяющую, правильно ли в нём расставлены круглые скобки.

Листинг программы:

static void Main(string[] args){

Console.Write("Введите арифметическое выражение: ");

string expression = Console.ReadLine();

bool isBalanced = CheckBrackets(expression);

if (isBalanced){

Console.WriteLine("Скобки в выражении расставлены правильно.");}

else{Console.WriteLine("Скобки в выражении расставлены неправильно.");}}

static bool CheckBrackets(string expression){ Stack<char> stack = new Stack<char>();

foreach (char ch in expression){if (ch == '('){stack.Push(ch);}

else if (ch == ')'){

if (stack.Count == 0){return false; } else {

stack.Pop(); }} } return (stack.Count == 0);}

Таблица 6.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| (1+(1+1)+1)+1( | Скобки в выражении расставлены неправильно. |

Источник: собственная разработка

Анализ результатов:

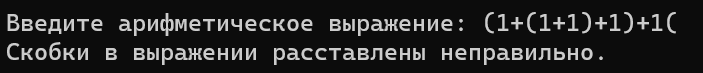


Рисунок 6.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Исключите из строки все заглавные символы.

Листинг программы:

static void Main(){

Console.WriteLine("Введите строку:"); string input = Console.ReadLine();

string output = RemoveUpperCase(input); Console.WriteLine("Результат:");

Console.WriteLine(output); }

static string RemoveUpperCase(string input)

{string output = ""; foreach (char c in input){if (!char.IsUpper(c)){output += c;}}

return output;}

Таблица 6.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| ЯсЯЧлСиРТшкРАом мнАоАго коАнАфеАт | Результат:  слишком много конфет |

Источник: собственная разработка

Анализ результатов:

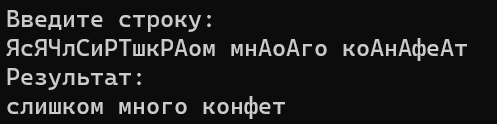


Рисунок 6.2 – Результат работы программы

Источник: собственная разработка

Задание 3. С клавиатуры вводится предложение. Результаты всех действий

вывести на экран. Поменять местами первое и последнее слова в предложении. Склеить второе и третье слова в предложении. Третье слово предложения вывести в обратном порядке. В первом слове предложения вырезать первые две буквы.

Листинг программы:

Console.Write("Введите предложение: "); string sentence = Console.ReadLine();

string sentence1 = sentence; string sentence2 = sentence;

string sentence3 = sentence; string sentence4 = sentence;

string[] words = sentence1.Split(' '); string temp = words[0];

words[0] = words[words.Length - 1]; words[words.Length - 1] = temp;

sentence1 = string.Join(" ", words);

Console.WriteLine("1.Переставим первым и последним словами: "); Console.WriteLine(sentence1);

words = sentence2.Split(' ');

words[1] += words[2];

for (int i = 2; i < words.Length - 1; i++){words[i] = words[i + 1];}

Array.Resize(ref words, words.Length - 1);

sentence2 = string.Join(" ", words); Console.WriteLine("2.Объединим вторым и третьим словами: "); Console.WriteLine(sentence2);

words = sentence3.Split(' ');

char[] chars = words[2].ToCharArray();

Array.Reverse(chars); string reversed = new string(chars);

Console.WriteLine("3.Третье слово в обратном порядке: "); Console.WriteLine(sentence3);

words = sentence4.Split(' ');

words[0] = words[0].Substring(2);

sentence4 = string.Join(" ", words);

Console.WriteLine("4.Первое слово без первых двух букв: "); Console.WriteLine(sentence4);

Таблица 6.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Сегодня тарелку мишки разбил его брат, другой мишка | 1.Переставим первым и последним словами:  мишка тарелку мишки разбил его брат, другой Сегодня  2.Объединим вторым и третьим словами:  Сегодня тарелкумишки разбил его брат, другой мишка  3.Третье слово в обратном порядке:  Сегодня тарелку мишки разбил его брат, другой мишка  4.Первое слово без первых двух букв:  годня тарелку мишки разбил его брат, другой мишка |

Источник: собственная разработка

Анализ результатов:

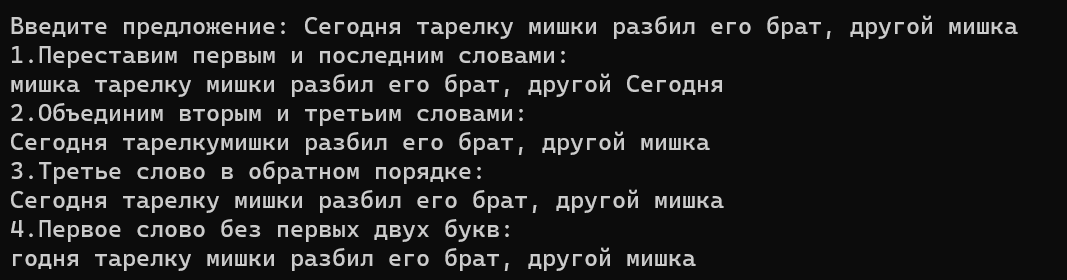


Рисунок 6.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Известны фамилия, имя и отчество пользователя. Найти его код личности. Правило получения кода личности: каждой букве ставится в соответствие число - порядковый номер буквы в алфавите. Эти числа складываются. Если полученная сумма не является однозначным числом, то цифры числа снова складываются и так до тех пор, пока не будет получено однозначное число.

Листинг программы:

const string Abc = @"абвгдеёжхийклмнопрстуфхцчшщъыьэюя";

Console.Write("Введите ФИО: ");

string[] fullName = Console.ReadLine().Split(' ');

int num = 0;

foreach (string word in fullName){

num += SumIndexInStr(word);}

int code = ToDetermineCodeNumber(num);

Console.WriteLine($"{code}");

int SumIndexInStr(string str) { int num = 0; string strLower = str.ToLower();

for (int i = 0; i < strLower.Length; i++)

{num += Abc.IndexOf(strLower[i]) + 1;}

return num;}

int ToDetermineCodeNumber(int num)

{int code = 0; while (num != 0)

{code += num % 10; num /= 10;if (code > 9 && num == 0){num = code;code = 0;}}return code;}

Таблица 6.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Александр Сергеевич Пушкин | 9 |

Источник: собственная разработка

Анализ результатов:

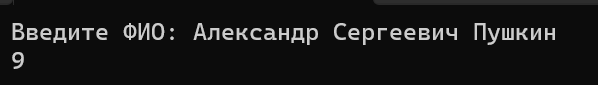


Рисунок 6.4 – Результат работы программы

Источник: собственная разработка